In the Claims

1. (Cancelled)

2. (Currently amended) A method according to claim [[1]] 24 wherein the ~~selected~~ first data file comprises compressed ~~data~~ components which [[is]] are decompressed when copied to the executable memory.

3. (Currently amended) A method according to claim 2 wherein the ~~selected data~~ first data file is decompressed as a whole when copied to the executable memory.

4. (Currently amended) A method according to claim ~~1 or~~ 2 or 24 wherein one ~~part~~ or more components of the ~~selected~~ first data file is/are copied to the executable memory independently of ~~another part~~ other components of the ~~selected~~ first data file.

5. (Currently amended) A method according to claim [[1]] 24 wherein the further data file comprises compressed data which is decompressed when selectively copied to the executable memory.

6. (Currently amended) A method according to claim 5 wherein the further data ~~comprises a plurality of components and~~ file is decompressed component by component when selectively copied to the executable memory.

7. (Currently amended) A method according to claim [[1]] 24 wherein the ~~selected data~~ first data file comprises core operating system data for the computing device.

8. (original) A method according to claim 7 wherein the core operating system data comprises program code for enabling boot-up of the computing device and access to read only file system (ROFS) data for the computing device.

9. (Currently amended) A method according to claim 8 wherein the ~~selected~~ first data file further comprises selected components of the ~~read-only file system~~ ROFS data.

10. (Currently amended) A method according to claim [[1]] 24 wherein the further data file comprises ~~read-only file system~~ ROFS data.

11. (Currently amended) A method according to claim 10 wherein the further data file comprises an executable program.

12. (Currently amended) A method according to claim 10 or 11 wherein the further data file comprises a dynamic link library.

13. (Currently amended) A method according to claim [[1]] 24 wherein the ~~selected~~ first data file is in the form of one or more ROM images.

14. (Currently amended) A method according to claim [[1]] 24 wherein the location of at least one of the ~~selected data~~ first and ~~the~~ further data files within the non-executable memory is determined by reading an address from a section of the non-executable memory.

15. (Currently amended) A method according to claim [[1]] 24 wherein additional data is selectively copied to the executable memory in addition to the first and further data files in the composite ~~data~~ file system.

16. (original) A method according to claim 15 wherein the additional data is selectively copied to the composite data file system.

17. (original) A method according to claim 15 or 16 wherein the additional data comprises a language pack image.

18. (Currently amended) A method according to ~~any one of~~ claims 15 ~~to 17~~ or 16 wherein a common driver is used to selectively copy the first and further data ~~and the additional data~~ files to the executable memory.

19. (Currently amended) A method according to ~~any one of~~ claims 15 ~~to 18~~ or 16 wherein the ~~selected data, the~~ first and further data files, and the additional data, are stored in a section of the non-executable memory locked to a user.

20. (Currently amended) A method according to claim [[1]] 24 wherein the non-executable memory is selected to comprise NAND flash memory.

21. (Currently amended) A method according to claim [[1]] 24 wherein the executable memory is selected to comprise random access memory (RAM).

22. (Currently amended) A computing device programmed to operate according to ~~the~~ a method ~~of~~ as claimed in claim 24 ~~any one of claims 1 to 21~~.

23. (Currently amended) Computer software arranged to cause a computing device to operate according to ~~the~~ a method as claimed in claim 24 ~~of any one of claims 1 to 21~~.

24. (New) A method of operating a computing device, the method comprising:

storing first and further data files in non-executable memory of the computing device;

copying the first data file from the non-executable memory to executable memory; and

presenting the first data file in executable memory and the further data file in non-executable memory as an executable composite file system;

wherein the first data file is accessed from the executable memory, and the further data file is accessed by selectively copying one or more components of the further data file to the executable memory.